

RB14 - Notions de base en informatique

Architecture matérielle

Florence Zara

LIRIS - Université Lyon 1

<http://liris.cnrs.fr/florence.zara>

E-mail: florence.zara@liris.cnrs.fr

Anatomie d'un ordinateur ou architecture matérielle

Objectifs

- De quoi se compose un ordinateur ?
- Mode de fonctionnement
 - du matériel
 - du logiciel
- Comment un ordinateur fonctionne ?
 - quels sont ses composants ?
 - comment et où l'information est stockée ?
 - comment s'exécute une instruction ?
 - comment une information passe de la mémoire au processeur, et réciproquement ?
 - comment sont gérées les communications avec l'extérieur ?

Un ordinateur type



Un ordinateur type

Unité centrale



Souris

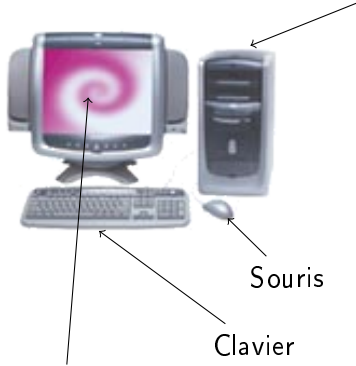
Clavier

Écran, hauts-parleur



Un ordinateur type

Unité centrale

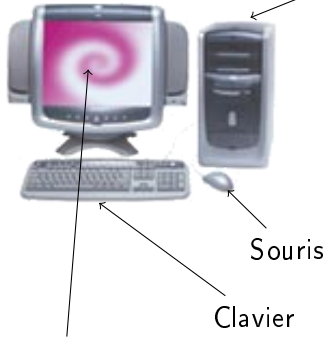


Écran, hauts-parleur

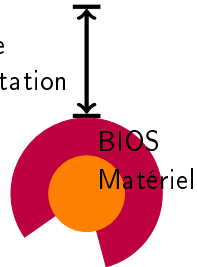


Un ordinateur type

Unité centrale

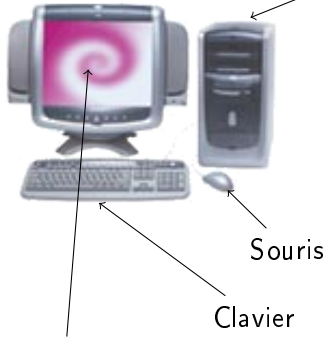


Système
d'exploitation



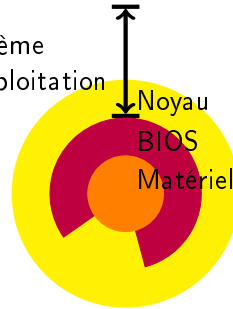
Un ordinateur type

Unité centrale



Écran, hauts-parleur

Système
d'exploitation



Un ordinateur type

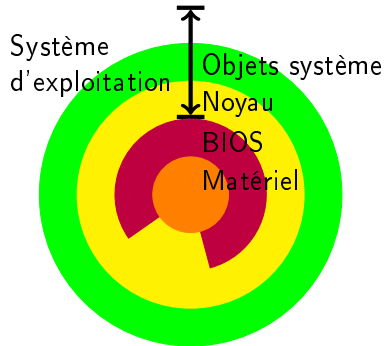
Unité centrale



Souris

Clavier

Écran, hauts-parleur



Un ordinateur type

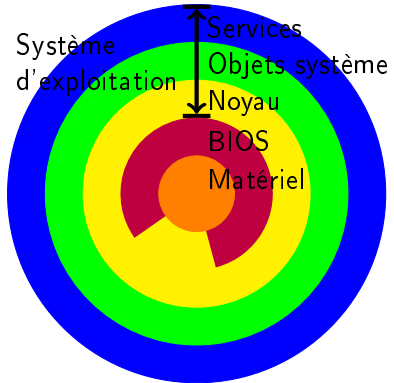
Unité centrale



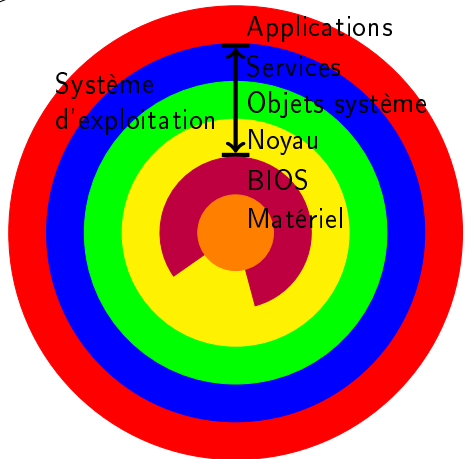
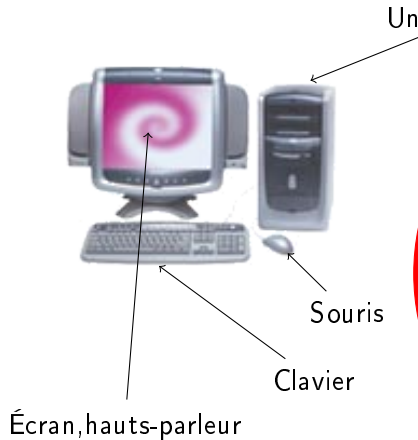
Souris

Clavier

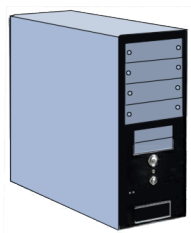
Écran, hauts-parleur



Un ordinateur type

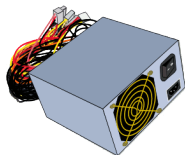


Le boîtier et l'alimentation



Caractéristiques du boîtier :

- dimensions
- nombre d'emplacements
- report d'Entrées/Sorties
- atténuation du bruit



Caractéristiques de l'alimentation :

- puissance délivrée
- nombre de branchements
- bruit

Branchement des composants sur la carte mère

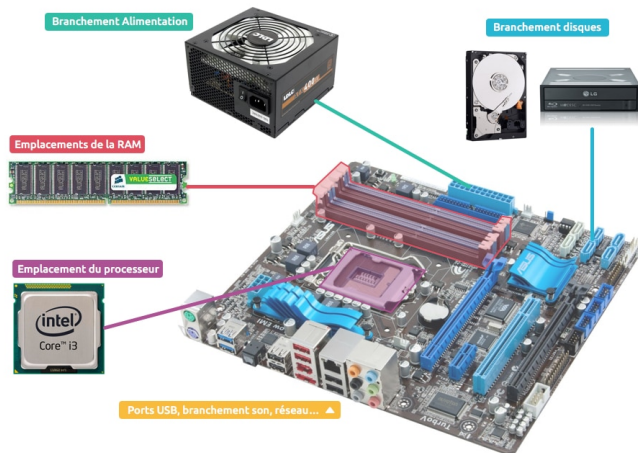
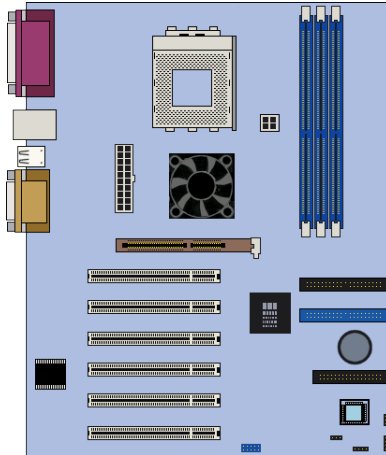
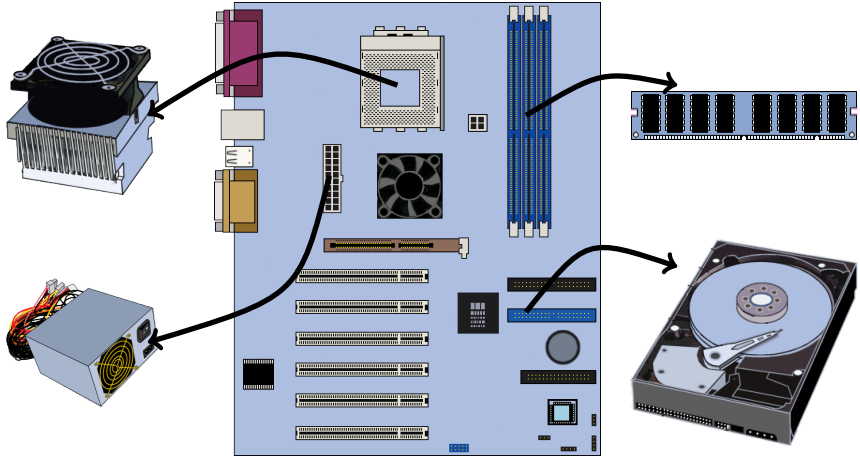


Image issue du site Xyoo

Branchement des composants sur la carte mère



Branchement des composants sur la carte mère



Branchement des composants sur la carte mère

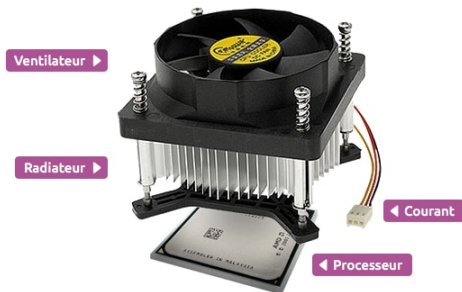
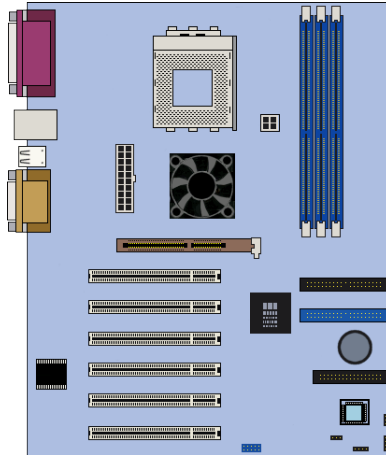
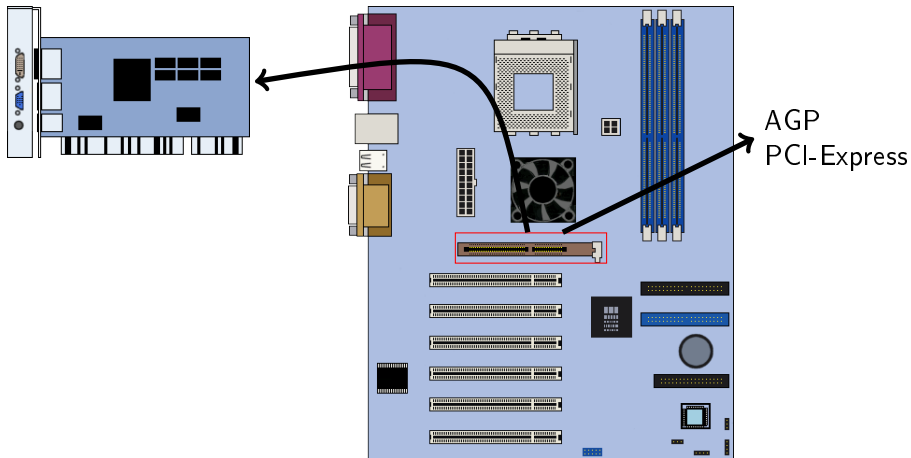


Image issue du site Xyoos

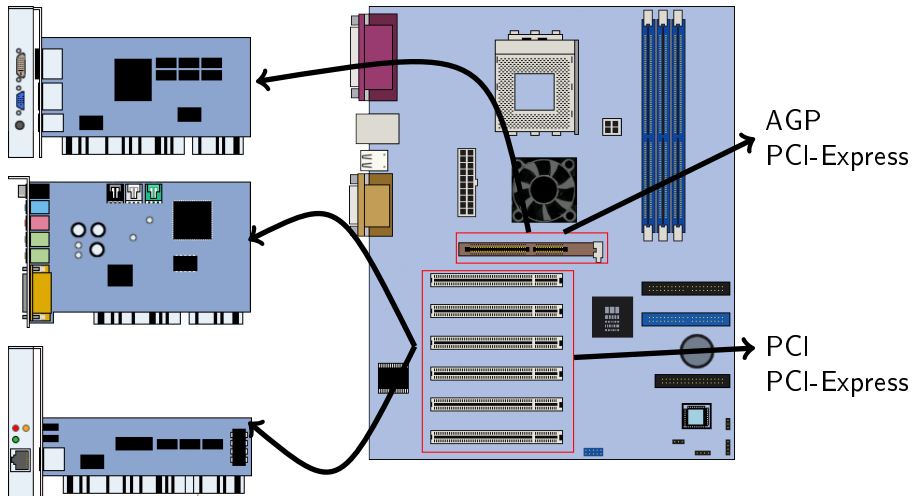
Emplacements des cartes d'extension sur la carte mère



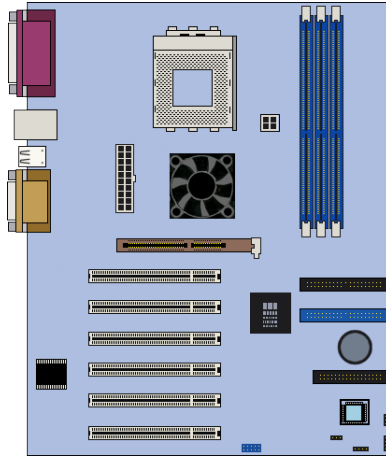
Emplacements des cartes d'extension sur la carte mère



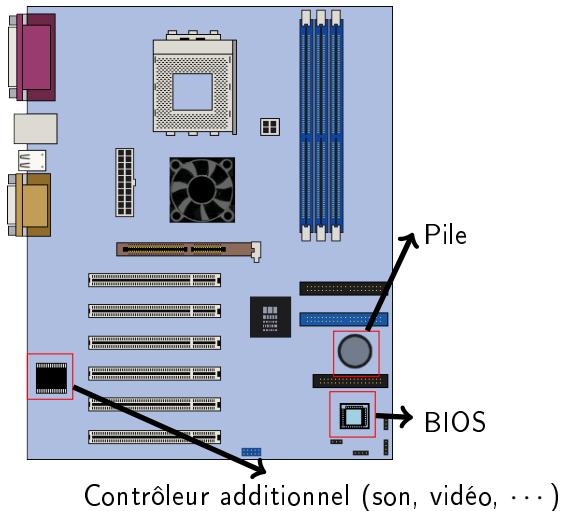
Emplacements des cartes d'extension sur la carte mère



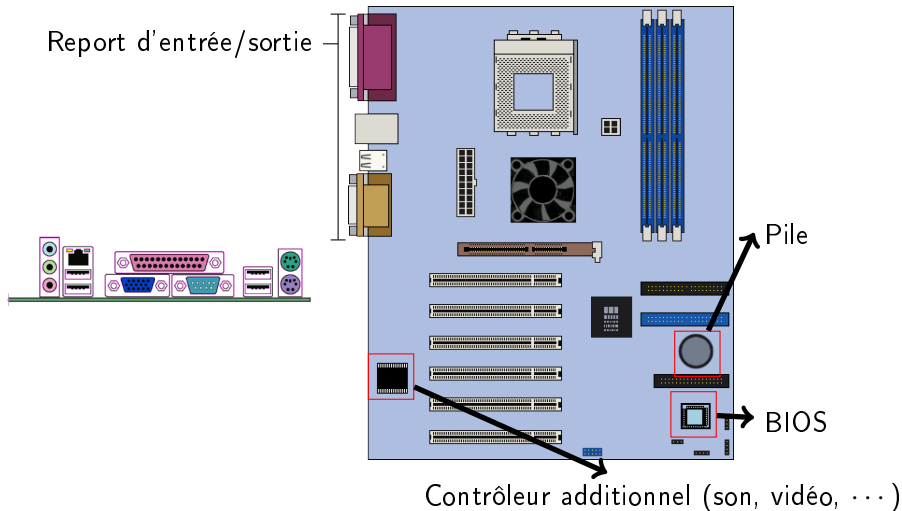
La carte mère



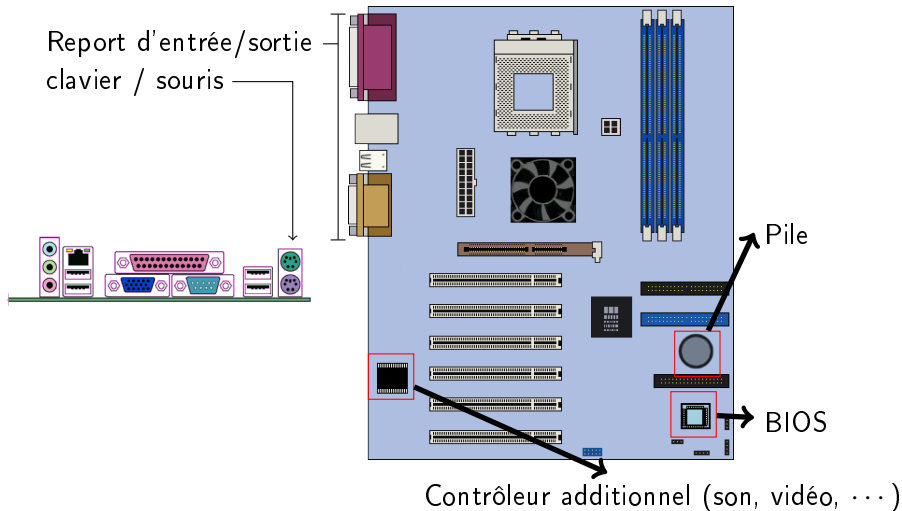
La carte mère



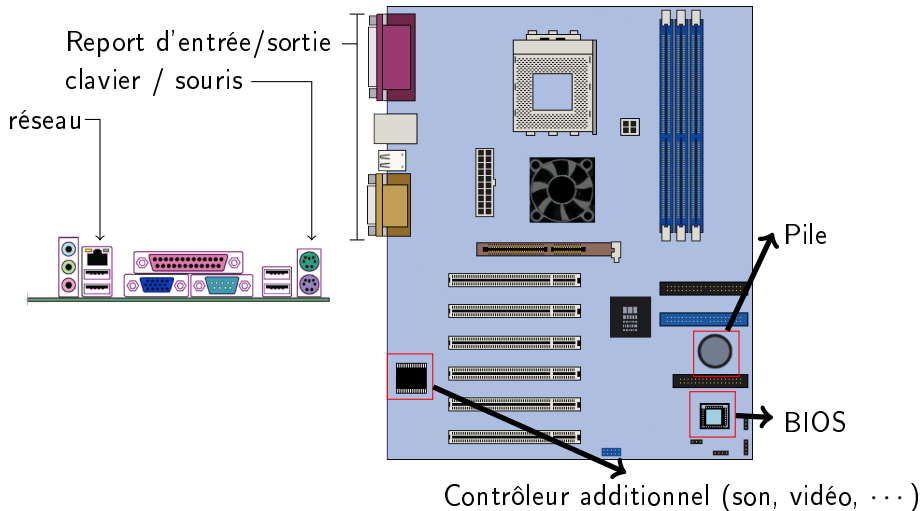
La carte mère



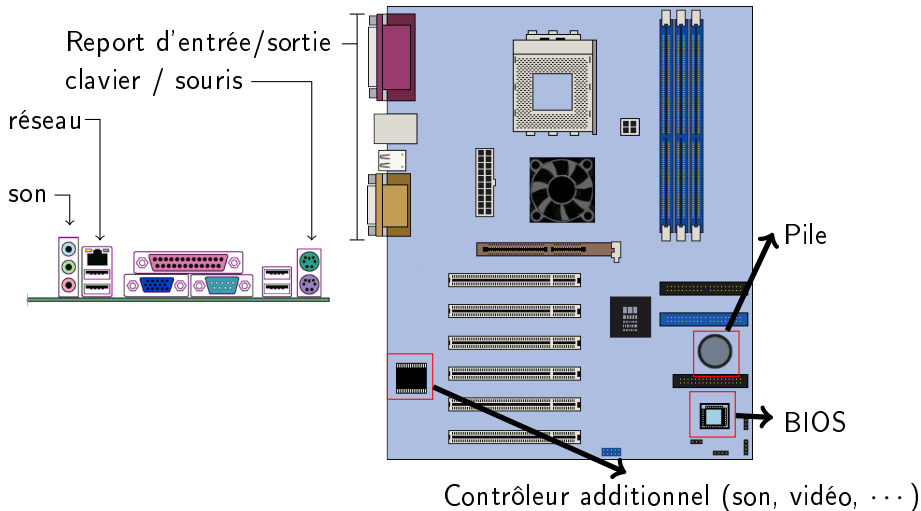
La carte mère



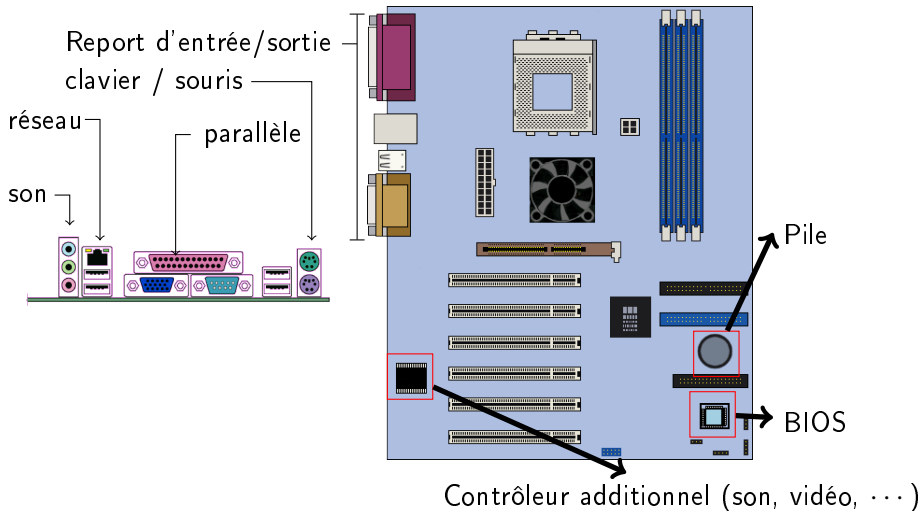
La carte mère



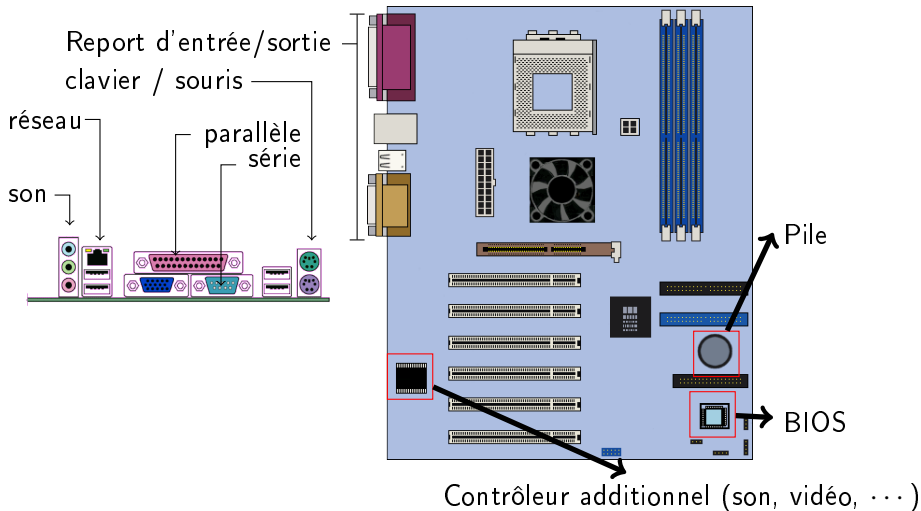
La carte mère



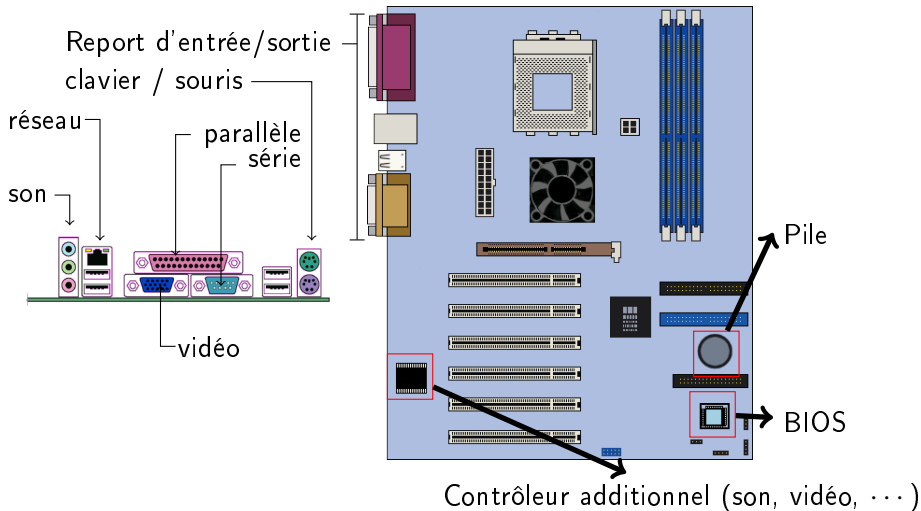
La carte mère



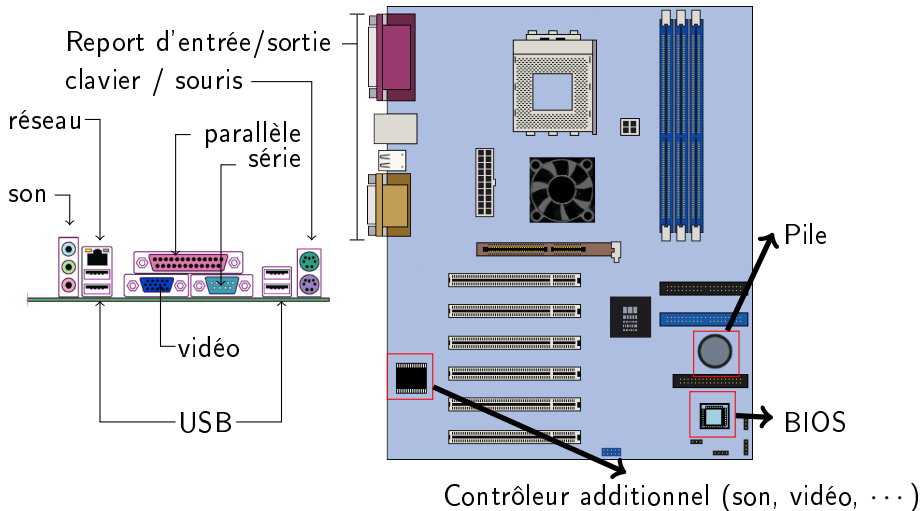
La carte mère



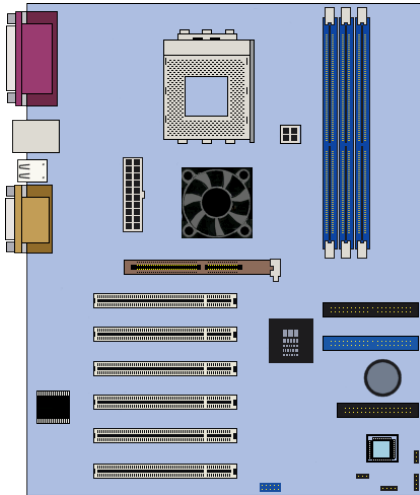
La carte mère



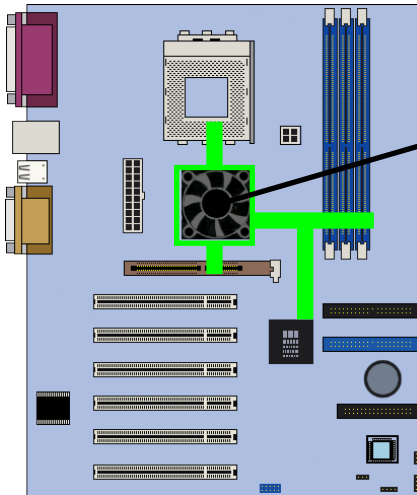
La carte mère



Les bus de communication de la carte mère

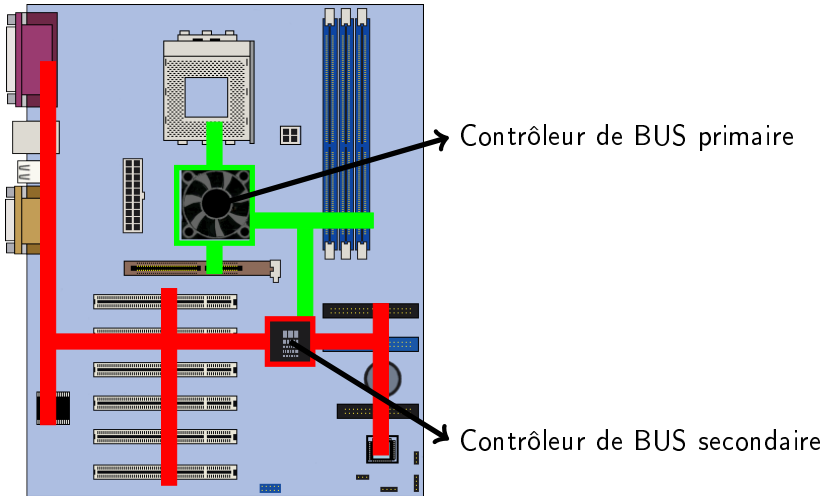


Les bus de communication de la carte mère

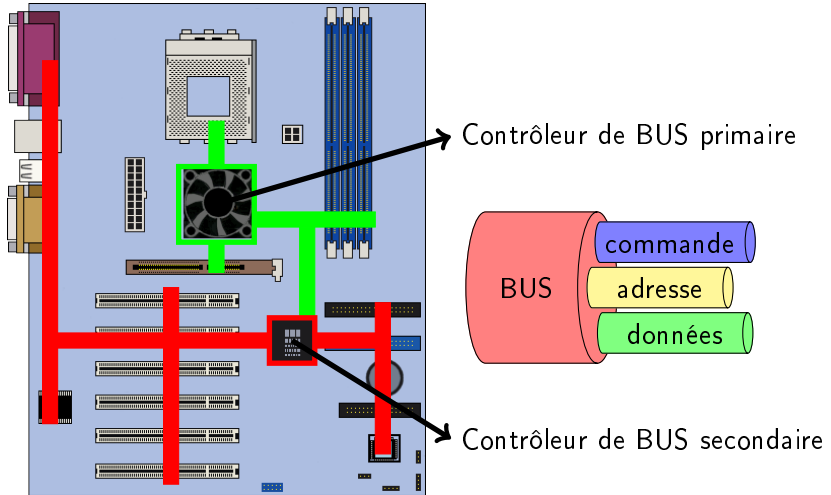


Contrôleur de BUS primaire

Les bus de communication de la carte mère



Les bus de communication de la carte mère



Caractéristiques de la carte mère

- Définition
 - circuit imprimé pour inter-connecter les composants d'un micro-ordinateur
- Le facteur d'encombrement
 - taille de la carte mère : ATX, microATX, ...
 - on ne peut pas mettre n'importe quelle carte mère dans n'importe quel boîtier
- Chipset
 - contrôleurs des communications sur les bus (fréquence, largeur)
 - intègrent souvent une carte vidéo, son et/ou réseau
- Type de support de processeur
 - ne peut pas mettre n'importe quel processeur sur n'importe quelle carte mère
- Les connecteurs d'entrée-sortie
 - nombre d'emplacement pour les extensions
 - nombre de ports USB, ...

La couche BIOS

- BIOS
 - Basic Input/Output System
 - système rudimentaire d'Entrée/Sortie
 - sur une puce de type EPROM
- Le BIOS sert
 - à démarrer l'ordinateur
 - à gérer la configuration matérielle au démarrage
 - de passerelle entre le matériel et certains systèmes d'exploitation

La couche BIOS

- Rôle du BIOS au démarrage
 - 1 vérifier que la configuration matérielle est conforme
 - 2 détecter le(s) disque(s)
 - 3 vérifier la mémoire
 - 4 lancer un système d'exploitation installé sur l'ordinateur
- Remarques
 - certains BIOS testent la température du processeur, la vitesse des ventilateurs, ...
 - les points 1 à 3 sont interruptibles (pour accéder à une interface de configuration « manuelle »)

Caractéristiques de la mémoire

- Capacité
 - quantité d'informations stockable
- Temps d'accès
 - temps entre la demande d'accès et l'accessibilité
- Temps de cycle
 - temps entre deux demandes d'accès
- Débit
 - quantité d'informations transférable par unité de temps
- Volatilité
 - persistance ou non de l'information lors de l'arrêt de l'ordinateur

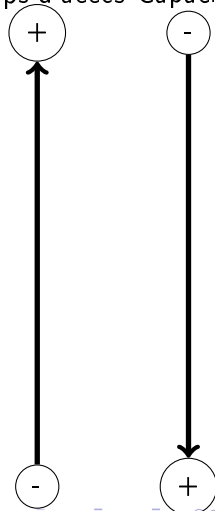
Types de mémoire

- Registres (dans le processeur, zone où sont stockées les chiffres)
- Mémoire cache (dans le processeur)
- Mémoire morte : ROM (read-only memory)
 - accessible en lecture seule
 - persistante
- Mémoire vive : RAM (random access memory)
 - accessible en écriture et en lecture
 - volatile
- Mémoire de masse : disque, ...

Types de mémoire

- Registres (dans le processeur, zone où sont stockées les chiffres)
- Mémoire cache (dans le processeur)
- Mémoire morte : ROM (read-only memory)
 - accessible en lecture seule
 - persistante
- Mémoire vive : RAM (random access memory)
 - accessible en écriture et en lecture
 - volatile
- Mémoire de masse : disque, ...

Temps d'accès Capacité



Codage des informations

Codage des données

- Codage des entiers : somme de puissances de 2
- Codage des caractères : code ASCII
- Codage des images : en fonction du format

Codage des programmes

- Programme
 - suite d'instructions élémentaires
- Instruction
 - opération élémentaire réalisée par le processeur
 - codée sur n octets
 - code de l'opération
 - code des opérandes

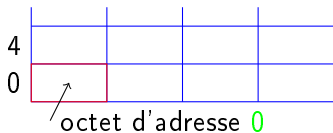
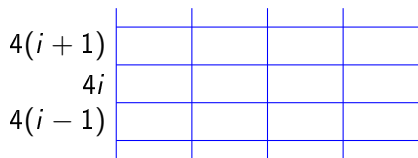
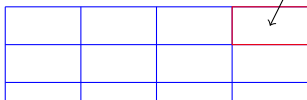
Codage des informations

Exemples de codage d'instruction

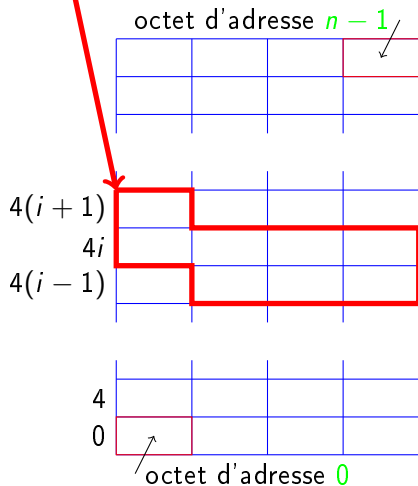
- ajouter 32 au contenu du registre RD
 - $RD \leftarrow RD + 32$ (algorithmme)
 - ADD RD, 32 (assembleur)
 - 0010 1100 00100000 (binaire)
- stocker, dans le registre RD, le contenu de la mémoire, à l'adresse 71
 - $RD \leftarrow Mem [71]$ (algorithmme)
 - MV< RD , Mem [71] (assembleur)
 - 0011 1100 01000111 (binaire)
- stocker en mémoire, à l'adresse 84, le contenu du registre RD
 - $RD \leftarrow Mem [84]$ (algorithmme)
 - MV> RD , Mem [84] (assembleur)
 - 0110 1100 01010100 (binaire)

Stockage de l'information

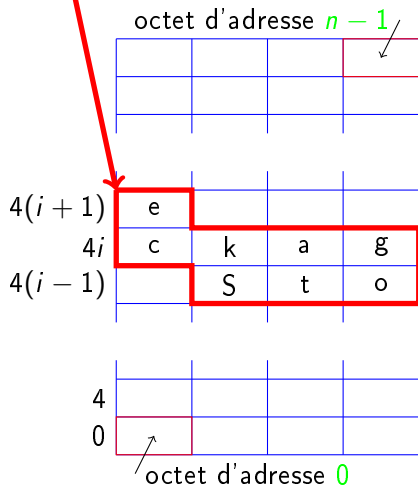
octet d'adresse $n - 1$



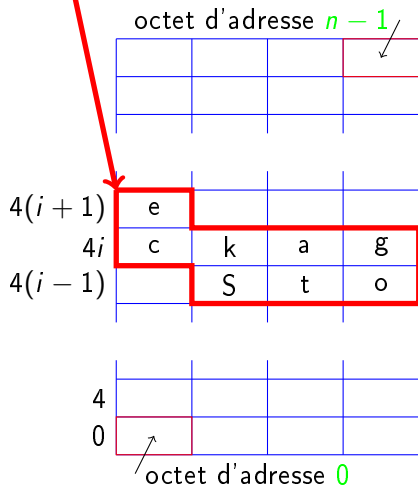
Stockage de l'information



Stockage de l'information

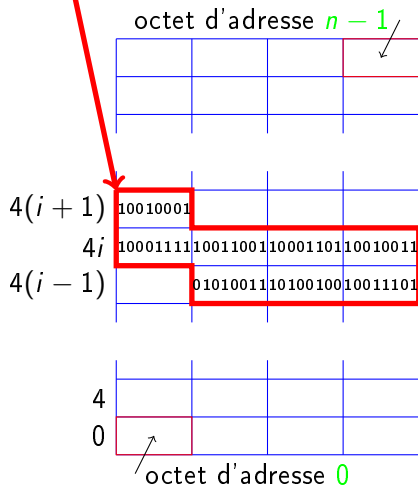


Stockage de l'information



lettre	ASCII	binaire
S	123	01010011
t	164	10100100
o	157	10011101
c	143	10001111
k	153	10011001
a	141	10001101
g	147	10010011
e	145	10010001

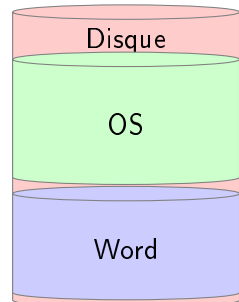
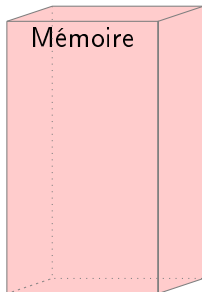
Stockage de l'information



lettre	ASCII	binaire
S	123	01010011
t	164	10100100
o	157	10011101
c	143	10001111
k	153	10011001
a	141	10001101
g	147	10010011
e	145	10010001

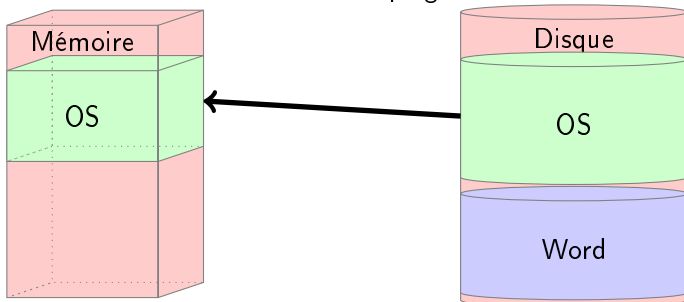
Principe général

- En mémoire
 - les programmes en cours d'exécution
 - les données sur lesquelles ils travaillent
- Gestionnaire de mémoire
 - alloue la mémoire pour charger le programme et ses données
 - gère les accès mémoire des différents programmes



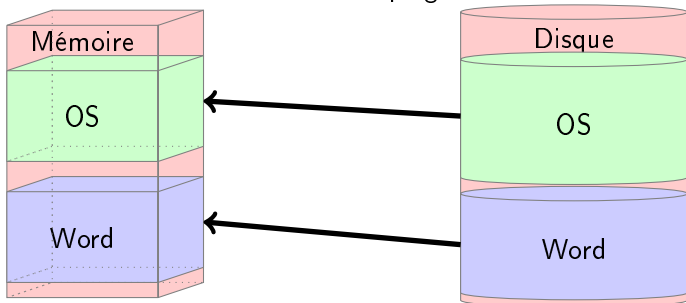
Principe général

- En mémoire
 - les programmes en cours d'exécution
 - les données sur lesquelles ils travaillent
- Gestionnaire de mémoire
 - alloue la mémoire pour charger le programme et ses données
 - gère les accès mémoire des différents programmes

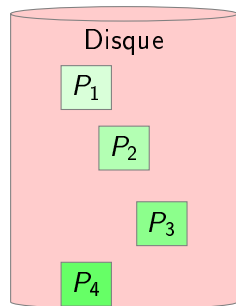


Principe général

- En mémoire
 - les programmes en cours d'exécution
 - les données sur lesquelles ils travaillent
- Gestionnaire de mémoire
 - alloue la mémoire pour charger le programme et ses données
 - gère les accès mémoire des différents programmes

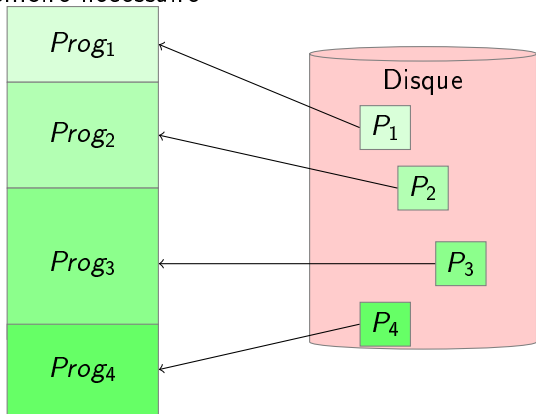


Mémoire virtuelle



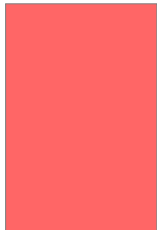
Mémoire virtuelle

Mémoire nécessaire

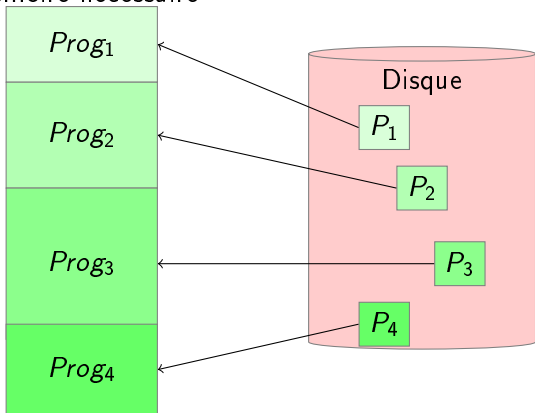


Mémoire virtuelle

Mémoire centrale

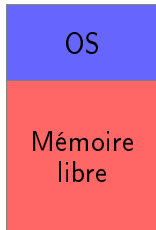


Mémoire nécessaire

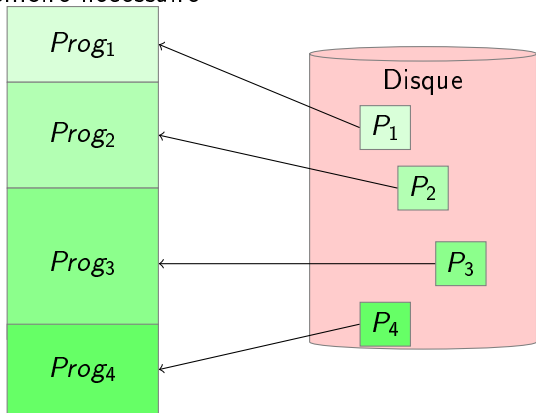


Mémoire virtuelle

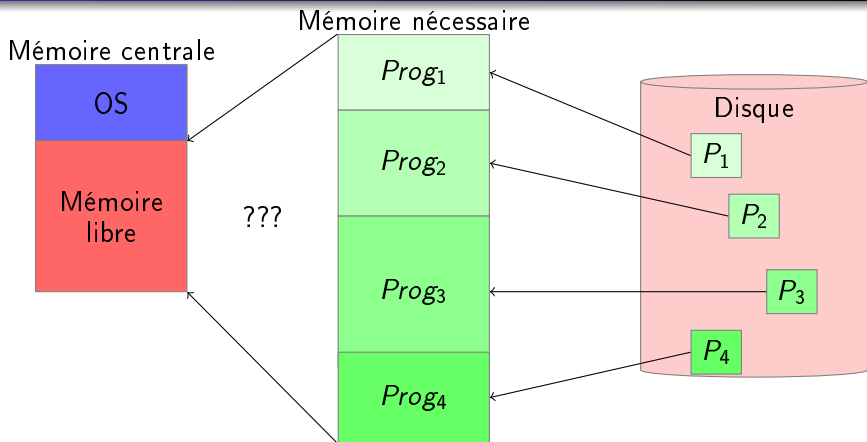
Mémoire centrale



Mémoire nécessaire



Mémoire virtuelle



- Comment faire pour attribuer plus de mémoire qu'il n'en existe physiquement ?

Principe

- Mémoire virtuelle
 - mémoire centrale (mémoire physique)?
 - une partie du disque dur allouée spécifiquement par le système (fichier d'échange : **swap**)
- Page mémoire
 - la mémoire virtuelle est découpée en pages
 - un programme en mémoire occupe un certain nombre de pages
 - à un moment donné
 - seules les pages nécessaires sont en mémoire centrale
 - les autres pages sont dans le fichier d'échange
 - elles seront échangées (swapped) au besoin

Exemple d'utilisation : état initial

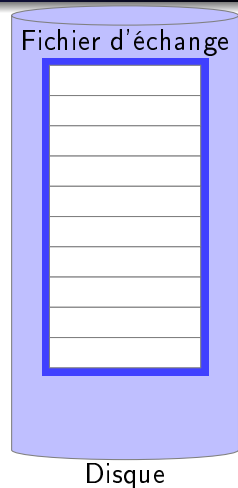
Programmes à charger



Mémoire centrale

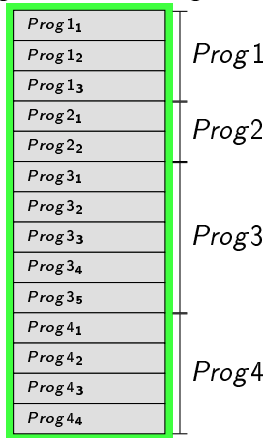


Fichier d'échange

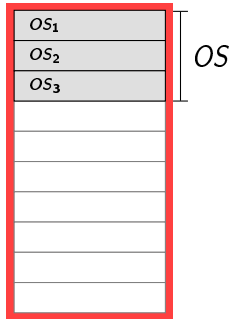


Exemple d'utilisation : état initial

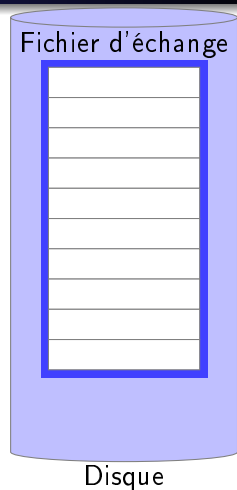
Programmes à charger



Mémoire centrale

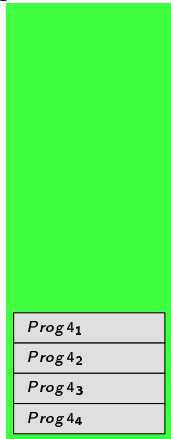


Fichier d'échange

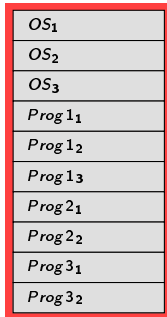


Exemple d'utilisation : exécution de *Prog4*

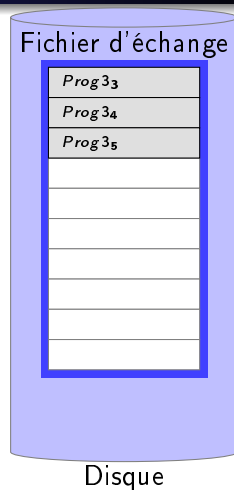
Programmes à charger



Mémoire centrale

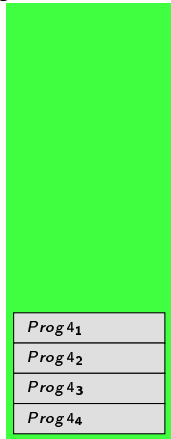


Fichier d'échange

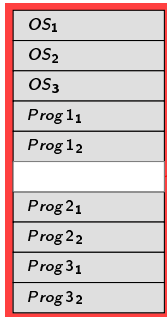


Exemple d'utilisation : exécution de *Prog4*

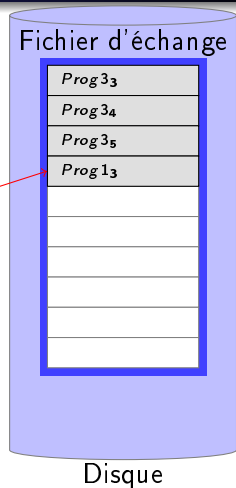
Programmes à charger



Mémoire centrale



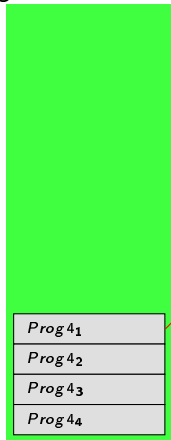
Fichier d'échange



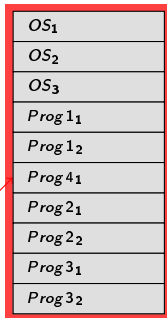
Disque

Exemple d'utilisation : exécution de *Prog4*

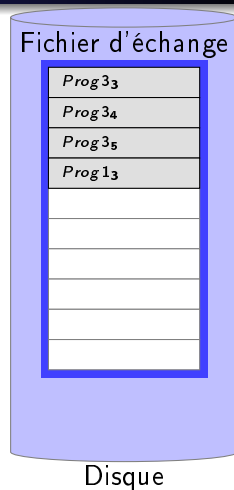
Programmes à charger



Mémoire centrale



Fichier d'échange

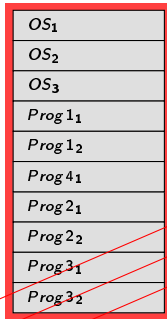


Exemple d'utilisation : exécution de *Prog4*

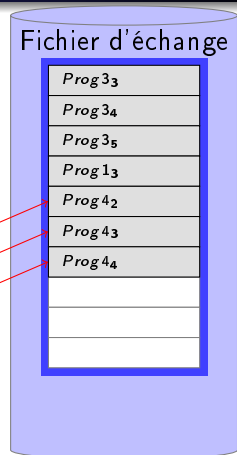
Programmes à charger



Mémoire centrale



Fichier d'échange



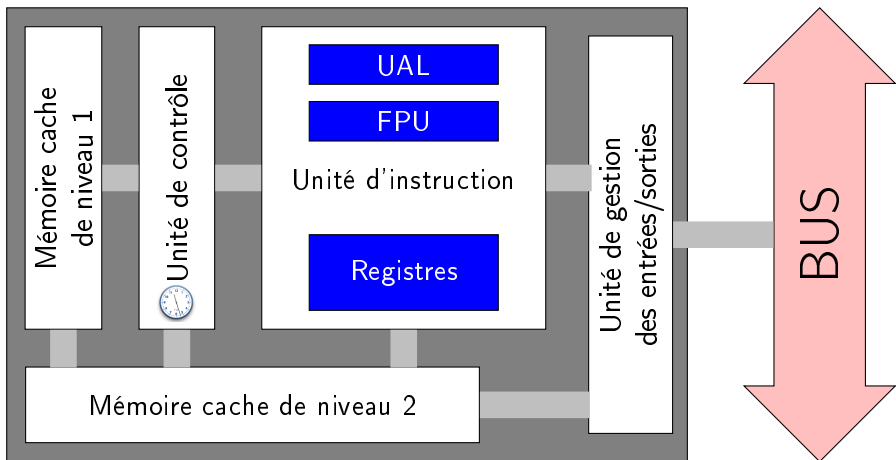
Disque

Rôle du processeur

- Processeur correspond au cerveau de l'ordinateur
- Il organise les échanges de données entre les composants (disque dur, mémoire RAM, carte graphique)
- Il fait les calculs (interaction avec utilisateur)
- Puissance du processeur exprimée en Hz
- Elle atteint les 3 Ghz (plusieurs milliards de calcul par seconde)
- De plus en plus d'ordinateurs équipés de plusieurs processeurs (car on a atteint la limite matériel pour faire des processeurs plus puissants)



Schéma simplifié du processeur



Le processeur

- L'unité de contrôle
 - l'horloge (qui séquence les instructions)
 - le pointeur de programme (adresse de l'instruction)
 - le registre d'instruction (codage de l'instruction)
 - le décodeur d'instruction
- L'unité d'instruction
 - registres (pour stocker des résultats temporaires)
 - unités de calcul
 - UAL : Unité Arithmétique et Logique
 - FPU : Floating-Point Unit
 - ...
- L'unité de gestion des E/S
- Les mémoires cache

Caractéristiques du processeur

- Fréquence de l'horloge
- Taille des différentes mémoires caches
- Architecture 16 bits, 32 bits, 64 bits ou 128 bits
- Technologies
 - MMX
 - SSE
 - Hyper-threading
 - Dual-core, multi-core

Principe

- Rappels
 - les instructions sont codées en binaire en mémoire
 - l'unité de contrôle sait récupérer le code d'une instruction et le décoder
- Avant l'exécution, le décodeur
 - connaît l'opération à réaliser
 - sait où se situent les opérandes
- Après l'exécution
 - les calculs/transferts sont réalisés sur les bons opérandes
 - le résultat a été récupéré au bon endroit
- Principe
 - récupérer les opérandes
 - transmettre les opérandes au composant qui va réaliser l'opération
 - indiquer à ce composant quoi faire

Opération arithmétique

Unité d'instruction

UAL

Registres

69

RD

RA

Unité de contrôle

Décodeur

Registres

00101100 00100000

RI

0x0000018A

PI



Opération arithmétique

Unité d'instruction

UAL

Registres

69

RD

RA

Unité de contrôle

Décodeur

Registres

00101100 00100000

RI

0x000018A

PI



Opération arithmétique

Unité d'instruction

UAL

Registres

69

RD

RA

Unité de contrôle

Décodeur

Registres

ADD RD, 32

00101100 00100000

RI



0x000018A

PI

Opération arithmétique

Unité d'instruction

UAL

Registres

69

RD

RA

Unité de contrôle

Décodeur

Registres

ADD RD, 32

00101100 00100000

RI

0x0000018A

PI

UAL : $\left\{ \begin{array}{l} op \leftarrow + \\ x \leftarrow RD \\ y \leftarrow 32 \\ x \ op \ y \rightarrow RD \end{array} \right.$



Opération arithmétique

Unité d'instruction

UAL

32

+69

—

101

Registres

69

RD

RA

Unité de contrôle

Décodeur

ADD RD, 32



Registres

00101100 00100000

RI

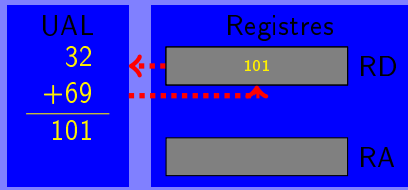
0x0000018A

PI

UAL : $\left\{ \begin{array}{l} op \leftarrow + \\ x \leftarrow RD \\ y \leftarrow 32 \\ x \ op \ y \rightarrow RD \end{array} \right.$

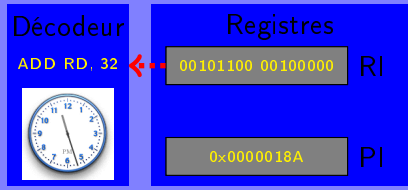
Opération arithmétique

Unité d'instruction



$$UAL : \left\{ \begin{array}{l} op \leftarrow + \\ x \leftarrow RD \\ y \leftarrow 32 \\ x \ op \ y \rightarrow RD \end{array} \right.$$

Unité de contrôle



Lecture depuis la mémoire

Unité d'instruction

Registres

RD

RA

Unité de contrôle

Décodeur



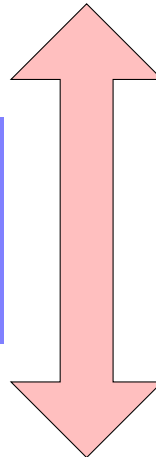
Registres

00101100 00100000 RI

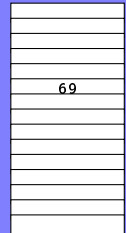
0x0000018A PI

Unité de gestion des Entrées/Sorties

BUS



Mémoire



Contrôleur

Lecture depuis la mémoire

Unité d'instruction

Registres

RD

RA

Unité de contrôle

Décodeur



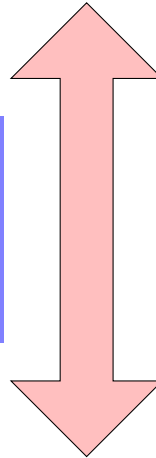
Registres

00101100 00100000 RI

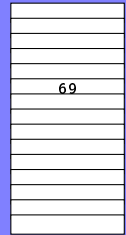
0x0000018A PI

Unité de gestion des Entrées/Sorties

BUS



Mémoire



Contrôleur

Lecture depuis la mémoire

Unité d'instruction

Registres

RD

RA

Unité de contrôle

Décodeur

$RD \leftarrow Mem[71]$



Registres

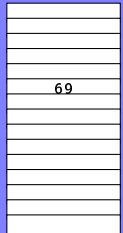
00101100 00100000 RI

0x0000018A PI

Unité de gestion des Entrées/Sorties

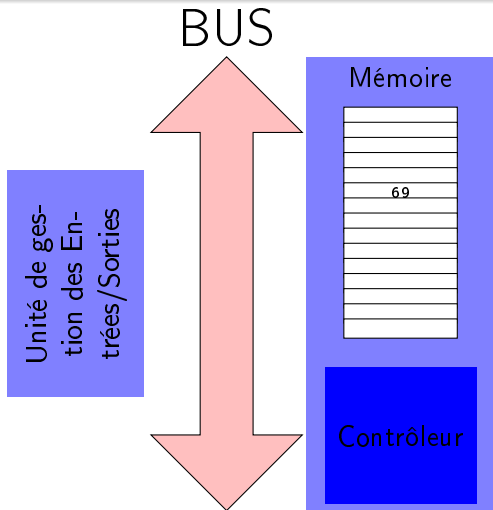
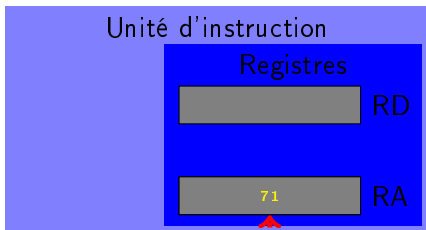
BUS

Mémoire

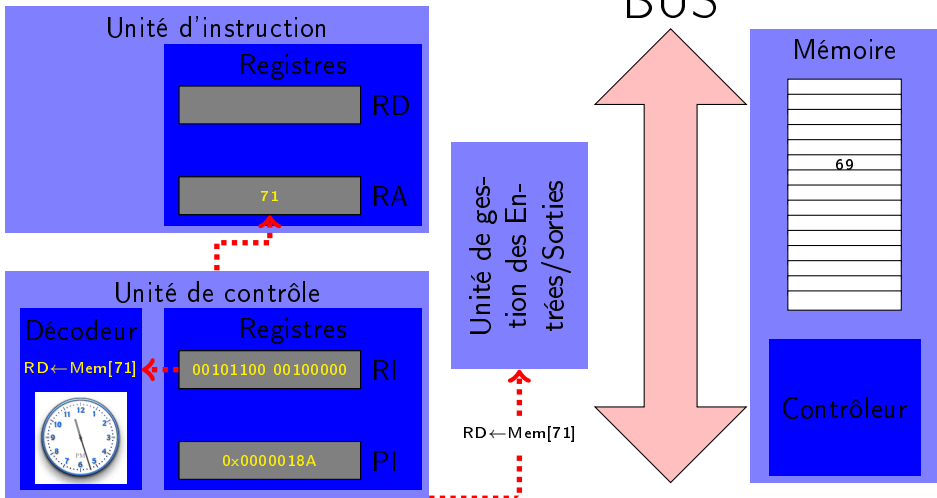


Contrôleur

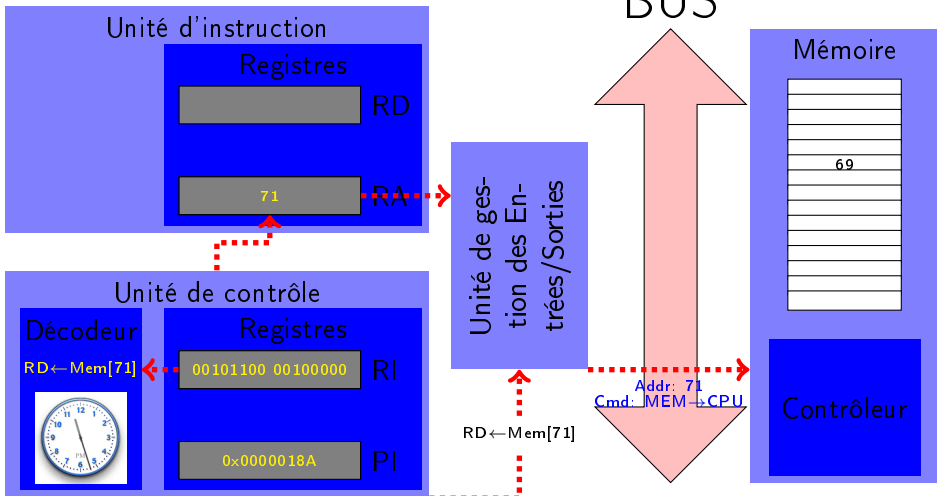
Lecture depuis la mémoire



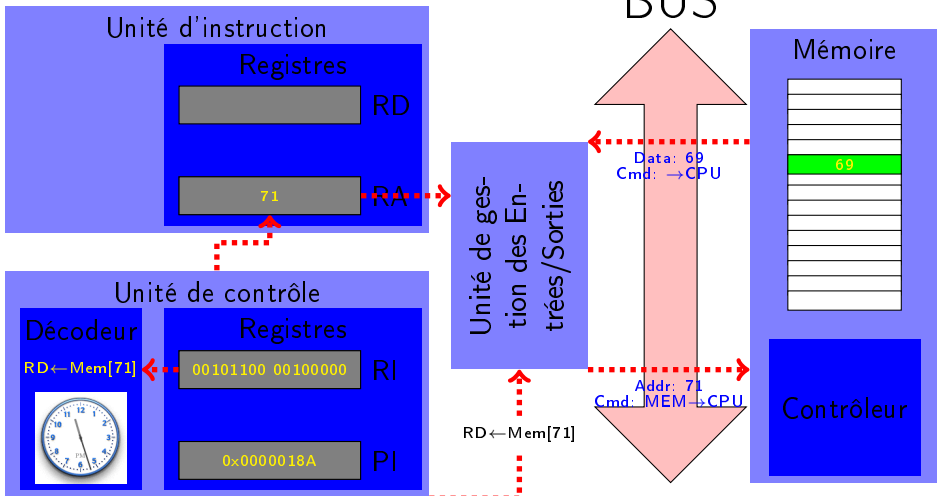
Lecture depuis la mémoire



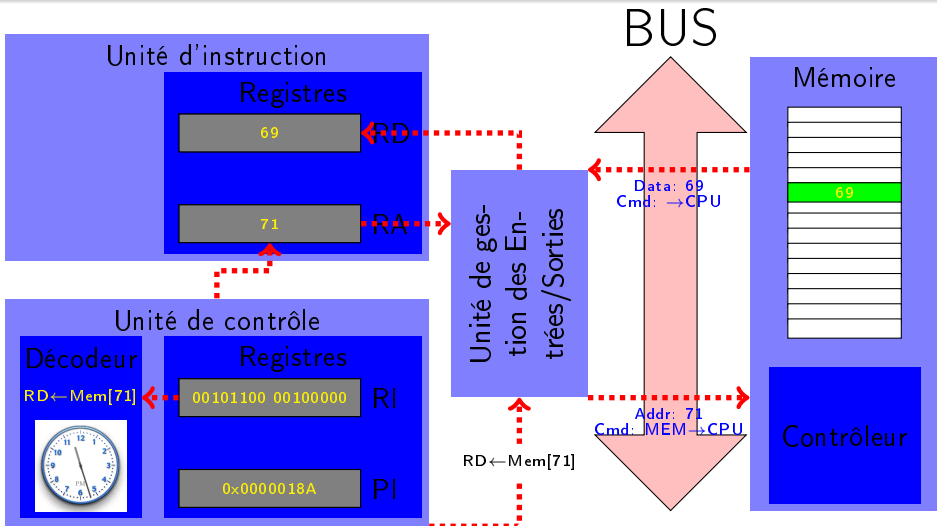
Lecture depuis la mémoire



Lecture depuis la mémoire



Lecture depuis la mémoire



Écriture dans la mémoire

Unité d'instruction

Registres

101

RD

RA

Unité de contrôle

Décodeur



Registres

00101100 00100000

RI

0x0000018A

PI

BUS

Unité de ges-
tion des En-
trées/Sorties

Mémoire

69

Contrôleur

Écriture dans la mémoire

Unité d'instruction

Registres

101 RD

RA

Unité de contrôle

Décodeur



Registres

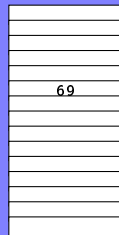
00101100 00100000 RI

0x0000018A PI

BUS

Unité de gestion des Entrées/Sorties

Mémoire



Contrôleur

Écriture dans la mémoire

Unité d'instruction

Registres

101 RD

RA

Unité de contrôle

Décodeur

RD → Mem[71]



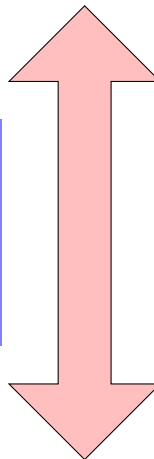
Registres

00101100 00100000 RI

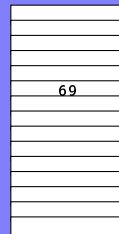
0x0000018A PI

Unité de gestion des Entrées/Sorties

BUS

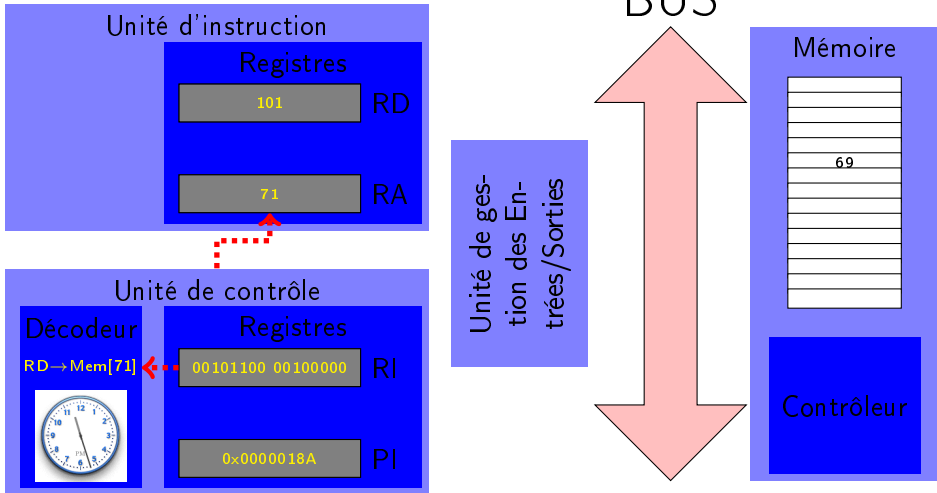


Mémoire

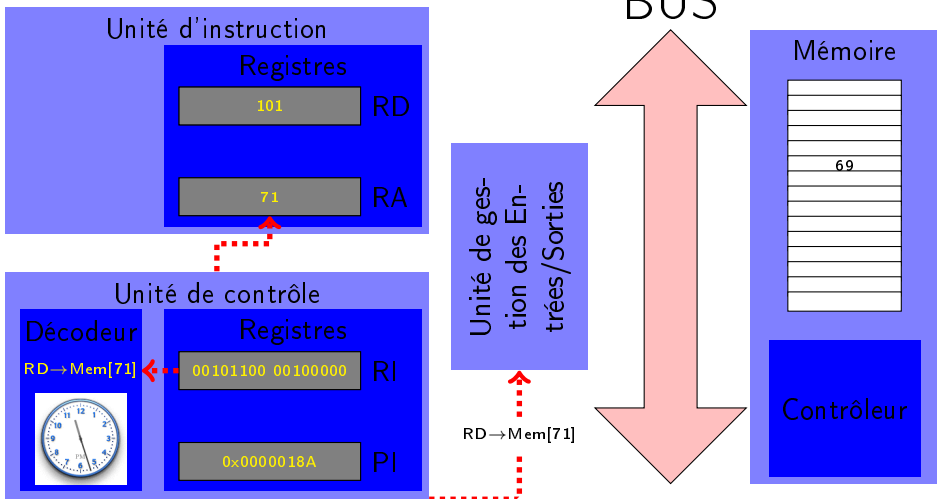


Contrôleur

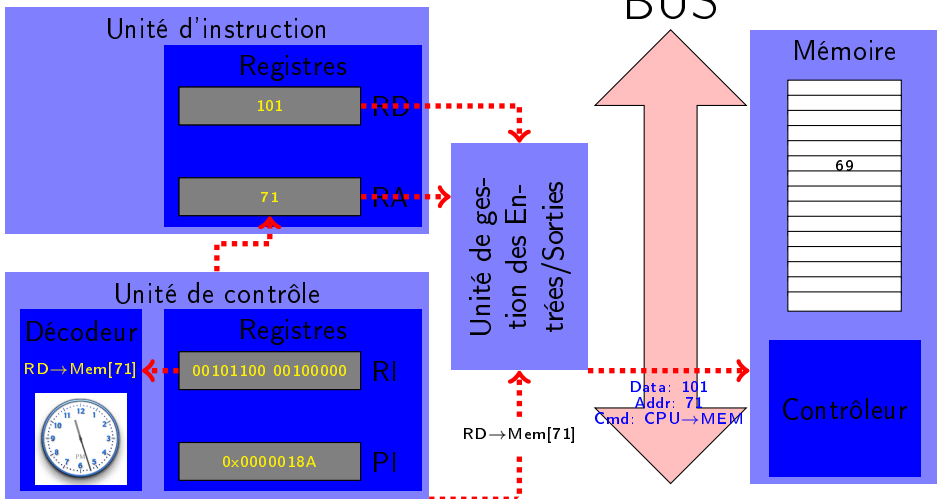
Écriture dans la mémoire



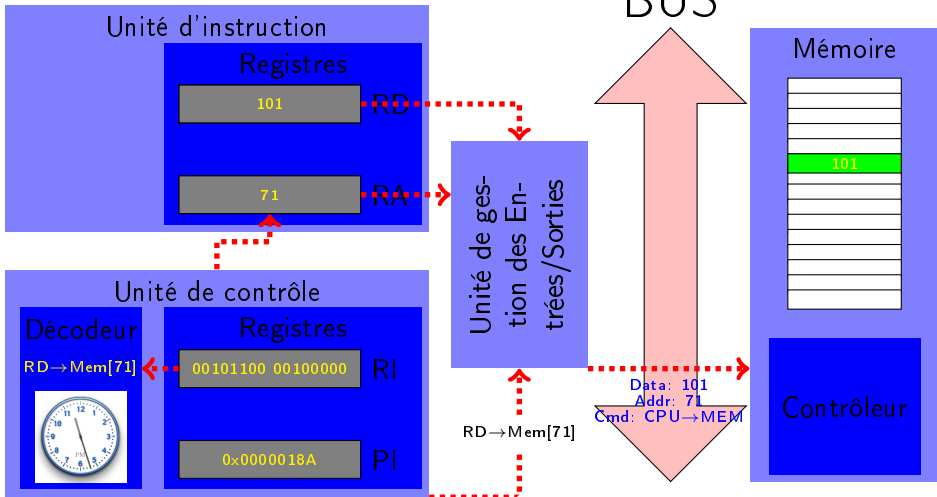
Écriture dans la mémoire



Écriture dans la mémoire



Écriture dans la mémoire



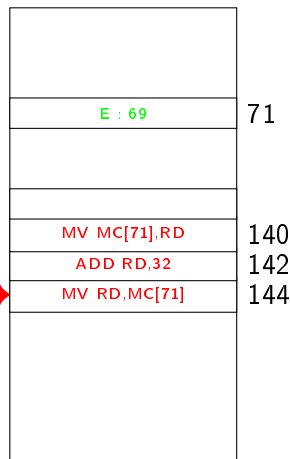
Principe

- "Modifier la casse" d'un caractère ($E \rightarrow e$)
- Rappel : codes ASCII pour E et e : 69, 101
- Solution : ajouter 32 au code ASCII du caractère
- Programme à exécuter
 - récupérer la première lettre du mot (à l'adresse 4i)
 - la convertir en minuscule
 - mettre la nouvelle "valeur" en mémoire au même endroit
- Instructions en mémoire
 - `MV RD, MC [4i]`
 - `ADD RD, 32`
 - `MV MC [4i], RD`

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1			
t_2			
t_3			
t_4			
t_5			
t_6			

Mémoire centrale



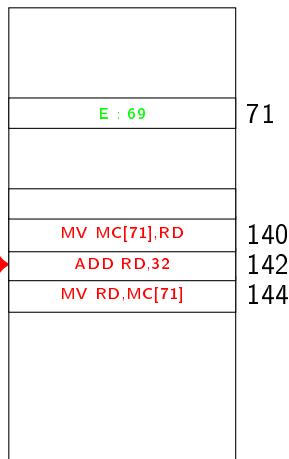
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2			
t_3			
t_4			
t_5			
t_6			

Mémoire centrale



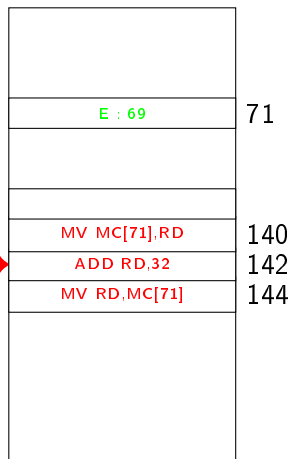
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2		E:69	71
t_3			
t_4			
t_5			
t_6			

Mémoire centrale



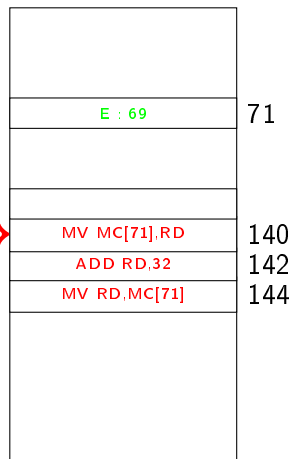
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2		E:69	71
t_3	ADD RD,32		
t_4			
t_5			
t_6			

Mémoire centrale



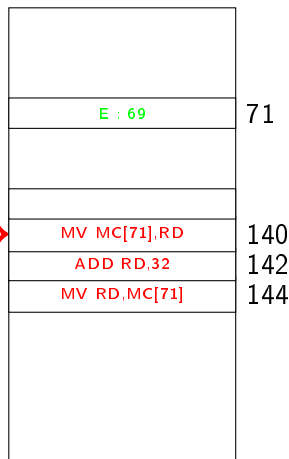
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2		E:69	71
t_3	ADD RD,32		
t_4		e:101	
t_5			
t_6			

Mémoire centrale



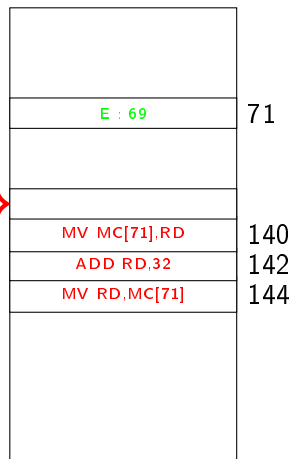
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2		E:69	71
t_3	ADD RD,32		
t_4		e:101	
t_5	MV MC[71],RD		
t_6			

Mémoire centrale



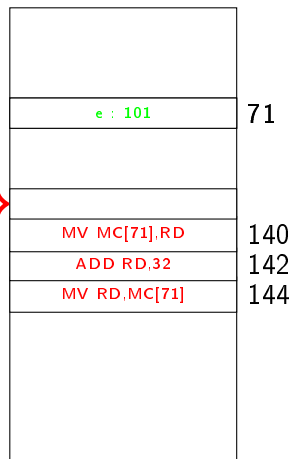
- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

PI

Exemple d'utilisation : état initial

temps	RI	RD	RA
t_0	???	???	???
t_1	MV RD,MC[71]		
t_2		E:69	71
t_3	ADD RD,32		
t_4		e:101	
t_5	MV MC[71],RD		
t_6			71

Mémoire centrale

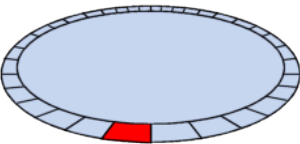
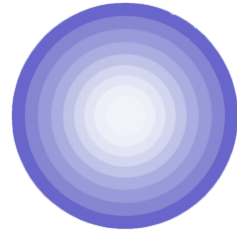
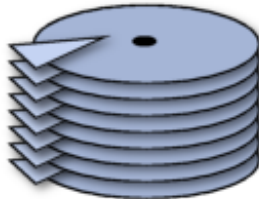


- À un instant t , l'horloge s'active
 - Si t est impair
 - Chargement de l'instruction dans RI et décodage
 - PI passe sur l'instruction suivante
 - Sinon exécution de l'instruction

Qu'est-ce qu'un processus ?

- Processus
 - partie d'un programme en cours d'exécution
 - un et un seul processus s'exécute à un moment donné
 - gestionnaire de processus
- Ordonnanceur
 - s'exécute à intervalles réguliers (de l'ordre de 100 ms) ou lors d'Entrées/Sorties
 - décide quel sera le processus suivant à exécuter
- Remarques
 - le changement de processus est rapide
 - les programmes semblent s'exécuter en parallèle

Exemple d'utilisation : état initial



- Un disque = un ensemble de plateaux empilés
- Un plateau = un ensemble de pistes concentriques
- Une piste = un ensemble de secteurs contigus
- Tous les secteurs ont une taille identique (en octets)
- Une adresse disque = (plateau, piste, secteur)

Caractéristiques d'un disque

- Capacité
- Taux de transfert
- Temps de latence
- Vitesse de rotation
- Temps d'accès moyen
- Interface de connexion
 - ATA/IDE débit maximum : 16 Mo/s
 - UDMA débit maximum : 133 Mo/s
 - SATAv1 débit maximum : 187 Mo/s
 - SATAv2 débit maximum : 375 Mo/s
 - SCSI débit maximum : 5 Mo/s à 640 Mo/s

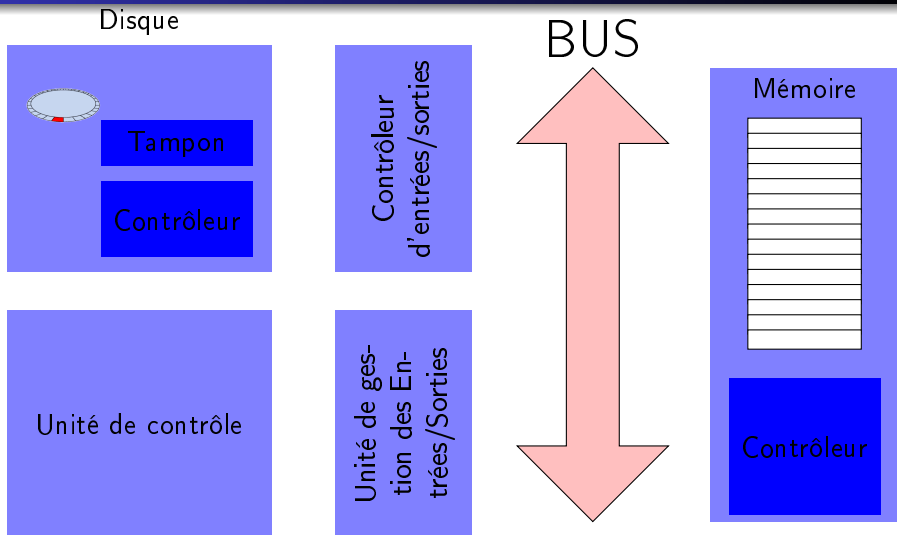
Maintenant technologie SSD remplace les disques durs

Évolution

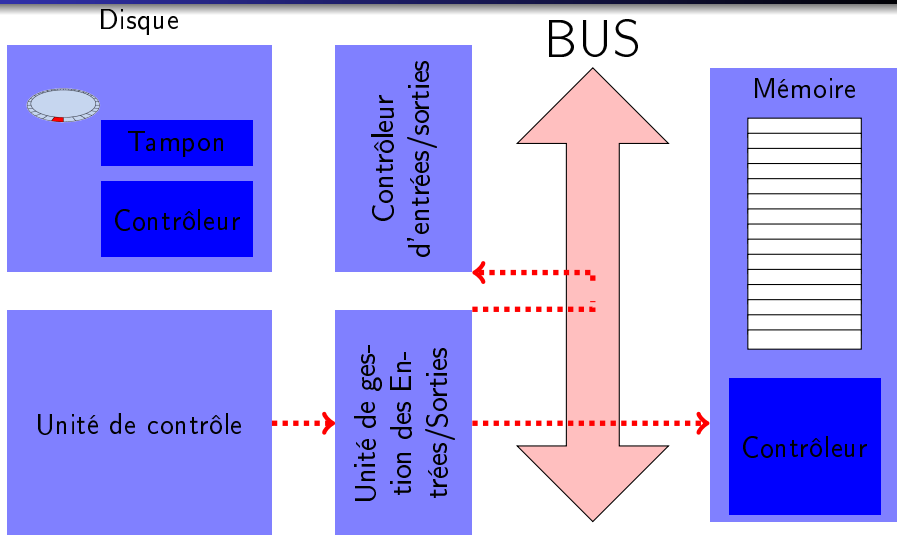
Les caractéristiques/prix évoluent très rapidement :

- en 2009, un SSD de 128 Go vaut environ 350 euros, ce qui reste nettement plus cher qu'un disque dur classique ;
- mi-2011, un SSD de 128 Go vaut moins de 200 euros ; la capacité des SSD dépasse désormais 1 To
- fin 2012, un SSD de 128 Go vaut environ 75 euros
- fin 2014, un SSD de 240 Go vaut environ 80 euros
- en 2017, un SSD de 1 To vaut moins de 300 euros
- en 2018, un SSD de 1 To vaut environ 190 euros
- en 2019, un SSD de 1 To vaut environ 90 euros

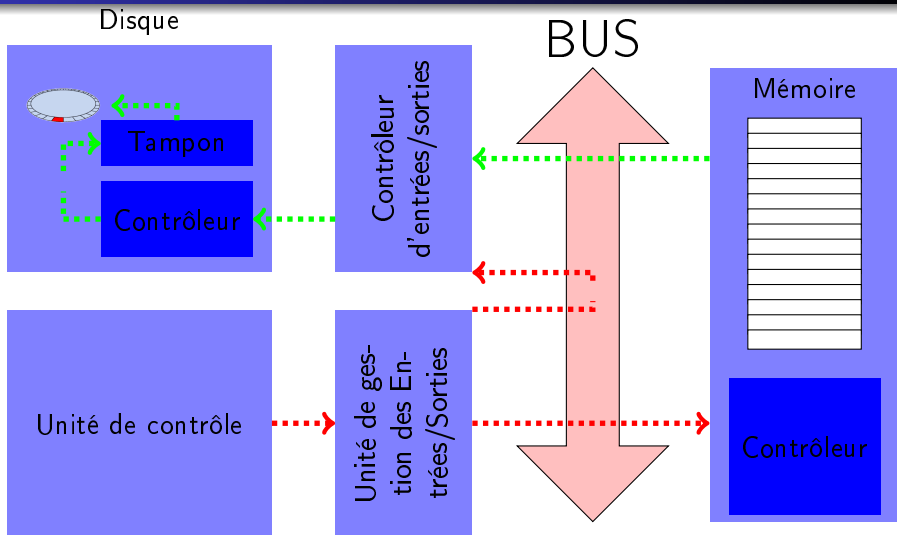
Sauvegarde sur le disque



Sauvegarde sur le disque

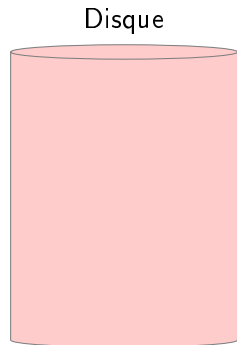


Sauvegarde sur le disque



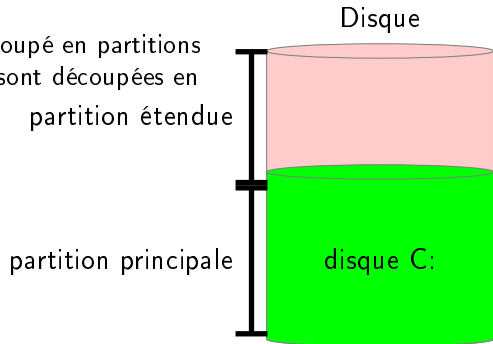
Principe

- Disque
 - périphérique (disque physique)
 - stockage des fichiers
 - découpé en blocs
- Découpage du disque
 - le disque physique est découpé en partitions
 - les partitions secondaires sont découpées en disques logiques



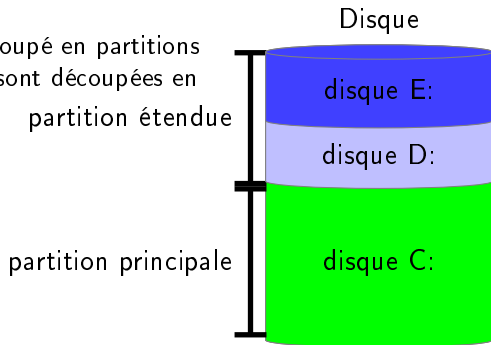
Principe

- Disque
 - périphérique (disque physique)
 - stockage des fichiers
 - découpé en blocs
- Découpage du disque
 - le disque physique est découpé en partitions
 - les partitions secondaires sont découpées en disques logiques



Principe

- Disque
 - périphérique (disque physique)
 - stockage des fichiers
 - découpé en blocs
- Découpage du disque
 - le disque physique est découpé en partitions
 - les partitions secondaires sont découpées en disques logiques

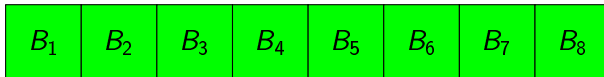


Gestionnaire de fichiers

- Système de gestion de fichiers
 - ensemble de programmes de manipulation de fichiers
 - spécifique au système d'exploitation
 - nommage
 - adressage sur le disque, ...
- Gestionnaire de fichiers
 - application permettant de manipuler les fichiers
 - il en existe plusieurs
 - gestionnaire simple
 - explorateur de fichiers
 - client ftp
 - ligne de commandes, ...

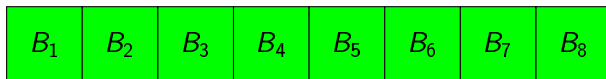
Localisation et structure interne des fichiers

- Localisation des fichiers sur disque
 - un fichier occupe un nombre entier de blocs
 - chaque bloc connaît le suivant
 - pour gérer l'espace disque
 - système de cartographie des blocs libres / occupés
- Structure interne d'un fichier



Localisation et structure interne des fichiers

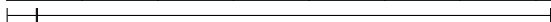
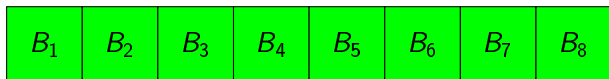
- Localisation des fichiers sur disque
 - un fichier occupe un nombre entier de blocs
 - chaque bloc connaît le suivant
 - pour gérer l'espace disque
 - système de cartographie des blocs libres / occupés
- Structure interne d'un fichier



En-tête du fichier : type (binaire, ASCII), taille utile, nb bloc, ...

Localisation et structure interne des fichiers

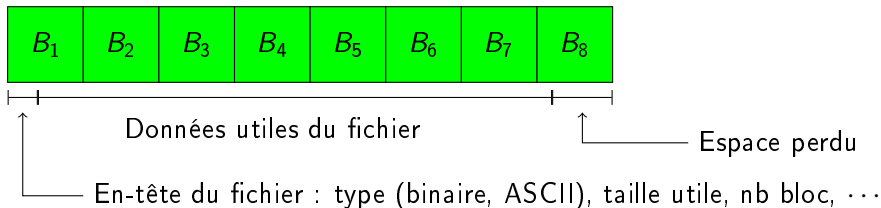
- Localisation des fichiers sur disque
 - un fichier occupe un nombre entier de blocs
 - chaque bloc connaît le suivant
 - pour gérer l'espace disque
 - système de cartographie des blocs libres / occupés
- Structure interne d'un fichier



En-tête du fichier : type (binaire, ASCII), taille utile, nb bloc, ...

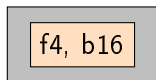
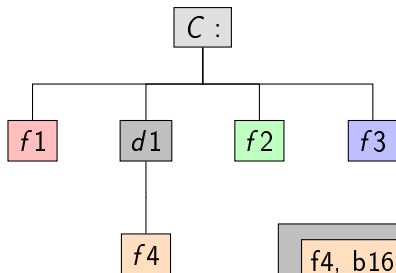
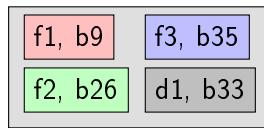
Localisation et structure interne des fichiers

- Localisation des fichiers sur disque
 - un fichier occupe un nombre entier de blocs
 - chaque bloc connaît le suivant
 - pour gérer l'espace disque
 - système de cartographie des blocs libres / occupés
- Structure interne d'un fichier



Structure interne des disques

- Les informations d'un dossier sont contenues dans un bloc
 - nom des objets contenus
 - bloc initial de ces objets
- Chaque bloc d'un fichier connaît son suivant



		C :					
	f1 ₁	f1 ₂	f1 ₃	f1 ₄	f1 ₅	f1 ₆	f1 ₇
f4 ₁	f4 ₂	f4 ₃	f4 ₄	f4 ₅	f4 ₆	f4 ₇	f4 ₈
		f2 ₁					
	d1		f3 ₁	f3 ₂	f3 ₂	f4 ₉	f4 ₁₀
f4 ₁₁	f4 ₁₂	f4 ₁₃	f4 ₁₄	f4 ₁₅			

Manipulation de fichiers sur le disque

- Création de fichiers
 - allocation du nombre de blocs nécessaires
 - inscription du lien dans le bloc du dossier
- Copie de fichiers
 - allocation du nombre de blocs nécessaires
 - copie des blocs du fichier vers les blocs cibles
 - inscription du lien dans le bloc du dossier
- Suppression de fichiers
 - les blocs sont marqués comme vides
 - suppression du lien dans le bloc du dossier
- Déplacement de fichiers sur un même disque
 - déplacement du lien entre les blocs des dossiers source et cible
- Déplacement de fichiers entre deux disques
 - copie du fichier depuis le disque source vers le disque cible
 - suppression du fichier sur le disque source

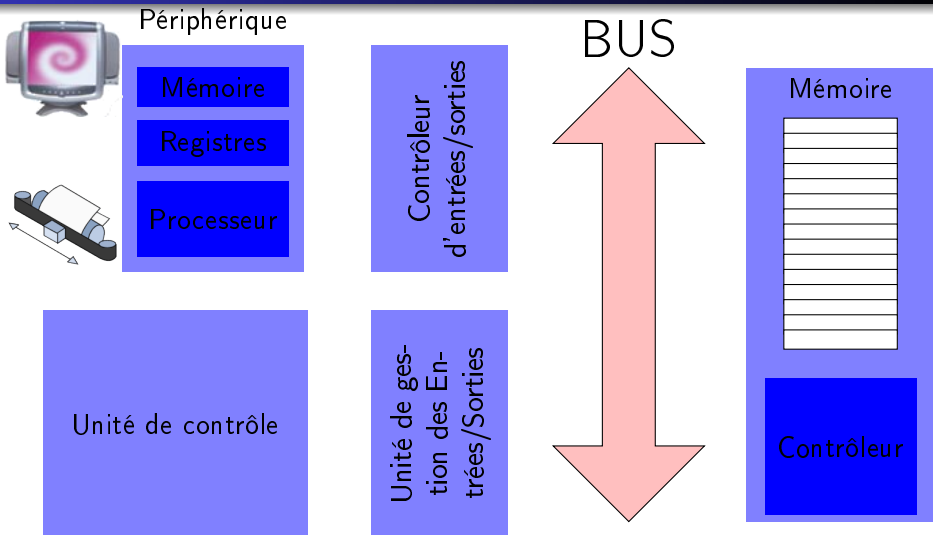
Vue d'ensemble

- Entrées/Sorties
 - entre le système d'exploitation
 - et les périphériques
- Gestion des Entrées/Sorties
 - communication entre un processus et les périphériques
 - adaptée à chaque périphérique
 - via le pilote du périphérique
- Pilote de périphérique
 - ensemble de fonctionnalités
 - échange d'informations
 - contrôle de l'état du périphérique
 - paramétrage

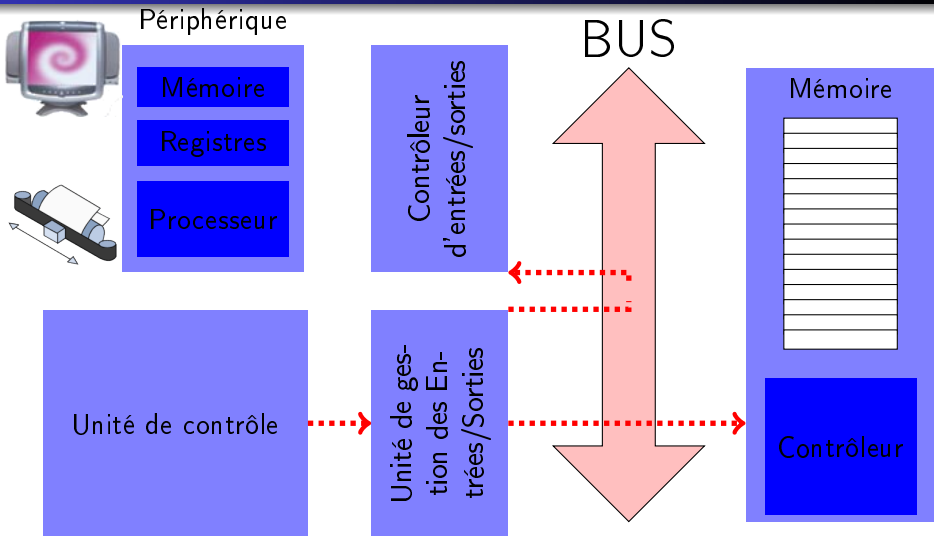
Exemples de pilotes (drivers)

- Pilotes « standard »
 - pilotes par défaut, fournis avec le système
 - disques
 - clavier, souris
 - affichage vidéo de base, ...
- Pilotes spécialisés
 - fournis avec le périphérique
 - imprimantes
 - scanners
 - cartes vidéo
 - souris non standard, ...

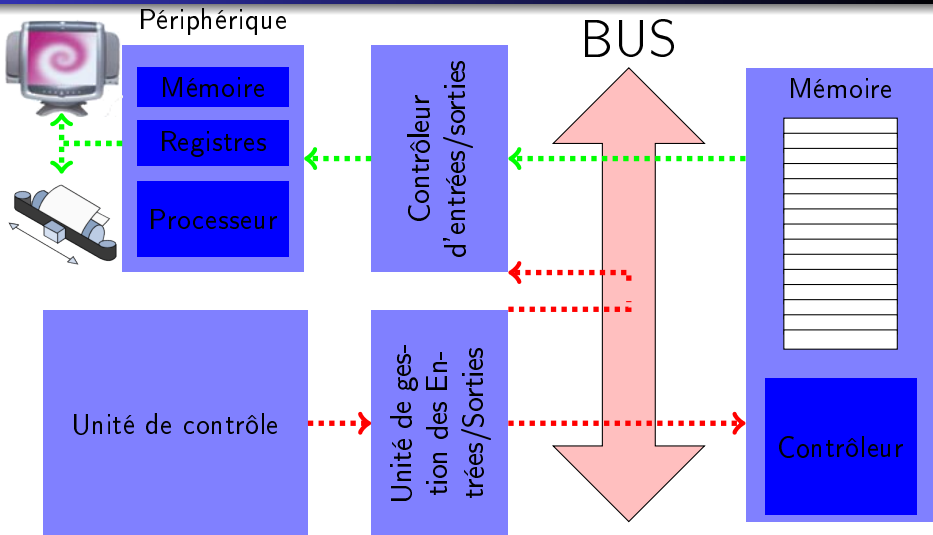
Communication avec le processeur



Communication avec le processeur



Communication avec le processeur



Caractéristiques écran/carte vidéo

- Carte vidéo
 - quantité de mémoire dédiée disponible
 - résolutions disponibles théoriquement
 - qualité différente selon les applications
 - capacités d'accélération 2D/3D
 - rapidité de rendu
 - bus de connexion
 - vitesse de transfert des données entre CPU et GPU
- Écran
 - résolutions utilisables
 - dimensions
 - temps de réponse
 - clarté/luminosité

Enchaînement de traitements

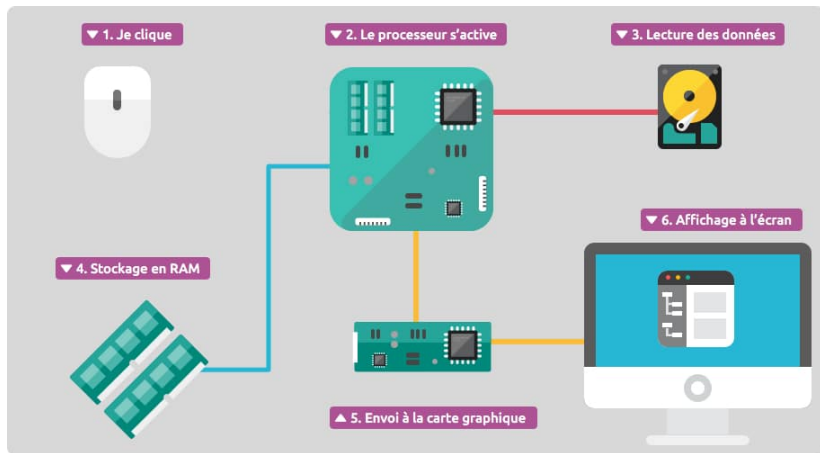


Image issue du site Xyoo

Objets système

- Entité "persistante" définie par :
 - un nom
 - des propriétés (modifiables)
- Connue du système d'exploitation
- Exemples
 - les comptes utilisateurs
 - les groupes d'utilisateurs
 - les disques
 - les fichiers
 - les périphériques accessibles (imprimantes, ...)
 - ...

Les services

- Ensemble de programmes permettant de gérer des ressources du système d'exploitation
 - comptes utilisateurs, accès réseau, site web, impressions, ...
- Installé comme composant complémentaire au système d'exploitation
- Disponible en permanence
- Rôle d'un service
 - utilisé par les applications pour réaliser certaines fonctions
 - exemple : service d'impression utilisé par un traitement de texte, un tableur, ...
- Un service est activé
 - à intervalles réguliers (horloge, vérification de mail, ...)
 - sur apparition d'un événement qu'il doit traiter (service d'impression, notificateur de mail, ...)